Encoding Information in Clustered-Dot Halftones

Robert Ulichney, Hewlett-Packard Co., Marlboro, MA
Matthew Gaubatz, Hewlett-Packard Co., Seattle, WA
Steven Simske, Hewlett-Packard Co., Ft. Collins, CO

Abstract

While barcodes are a popular means for encoding information for printed matter, they add unsightly overt content. If an image is already part of the composition of a printed label or page, hiding information in that image is an attractive alternative to barcodes. This paper offers a new method for encoding information in the halftone of an image. We focus on the class of techniques that perform clustered-dot halftoning, as commonly used in both dry toner and liquid toner electrophotographic processes. The method takes as input any grayscale image and a payload of data to be encoded and produces a bitonal clustered-dot halftone of that image with selected halftone clusters shifted to carry varying numbers of bits from the payload. The resulting data-bearing steganographic halftone is referred to as a "Stegatone". Because of the small size and large number of clustered-dot cells in printed halftones the bit density is quite high – over 2000 bytes/square-inch. Scans of test printed stegatones from a number of printers support the robustness of the method with high recovery rates.

Introduction

Technologies for enabling data-bearing hard copy afford a number of interesting applications. These include security and forensic applications for labels, packaging, signage, and documents in general. While barcodes are a popular means for encoding information, they add unsightly overt content. A more attractive approach is to embed information in images – not in the image file as is done in traditional watermarking, but in the halftone on the printed page. In this paper we present a solution for the class of rendering techniques used in most commercial printers: clustered-dot halftoning. Both dry and liquid electrophotographic printers use clustered-dots because they are more stable than dispersed-dot halftones. While dispersed-dot halftoning is preferred for inkjet printers, these printers can certainly also render clustered-dots, making this technique suitable for essentially all print products.

Hiding information in continuous-tone image data is often referred to as "watermarking" and has a long history of research. Since the nature of such encoded images change considerably when halftoned for printing, some work [1] has looked at methods that allow the continuous-tone encodings to survive the halftoning process. Most efforts to embed information in the halftone itself focus on dispersed-dot dithering applications. A survey [2] of such techniques was recently published. Most approaches use a form of error-diffusion.

One solution conveys data using blocks of output pixel shapes [3]. Some techniques employ a watermark to convey visual information [4][5]. One such idea uses two halftoned versions of the same image that must be overlaid to reveal the hidden bitonal watermark; the complementary halftones are called "conjugate pairs" [6][7]. Dispersed-dot dithering approaches also hide data by manipulating image edges [8], or by toggling pairs of pixels [9]. At Purdue, Allebach has pursued the policy of not disturbing the data and instead has focused on embedding data in sub-pixel offsets available in some electrophotographic printers; he calls this the "printer mechanism" in his feasibility studies [10].

Clustered-dot halftones have been used to carry information by creating asymmetric shapes in the clusters, such as ovals [11], and manipulating shape orientation to encode a bit. Limited information can be embedded in clustered-dot screens by altering their phase and frequency [12]. For recovering individual ink patterns from color clustered-dot printed halftones, a solution for separating the scan of such halftones is reported [13]. Anoto [14] covers an entire page with dots of the same size and shape where every dot is shifted from a nominal position as a form of encoding, but is not in any way used to halftone an image or encode an arbitrary payload.

There has been no known prior effort to use shifts of clustered-dot halftoned clusters as a means for embedding data.

Clustered-Dot Halftoning

The nature of any ordered dither is dictated by a deterministic, periodic array of threshold values. In the case of clustered-dot halftoning, the thresholds are arranged so that output pixels will form increasing sizes of white clusters as input values increase from full black, and then ever decreasing sizes of black clusters as input values further increase to full white. This rule or order of thresholds is first specified by a dither template as shown in Figure 1. This 8x8 matrix contains values from 0 to 63 that define the order that cells will be turned "on" or "white". This arrangement forms a classical 45-degree screen. On a 600 dpi printer, it will have a screen frequency of 106 lines/inch.

The 4x4 shaded regions depict shadow cells, and the 4x4 unshaded regions depict highlight cells. Shadow cells will be white holes surrounded by black, and highlight cells will be black clusters surrounded by white. The actual threshold values that will be used to compare against 8-bit input pixel values have to be normalized.

14	12	16	20	49	51	47	43
10	0	2	18	53	63	61	45
8	6	4	22	55	57	59	41
30	26	24	28	33	37	39	35
48	50	46	42	15	13	17	21
52	62	60	44	11	1	3	19
54	56	58	40	9	7	5	23
32	36	38	34	31	27	25	29

Figure 1. Dither Template.

To be mean-preserving, the follow relationship is used to scale the template values T[x,y] to threshold array values A[x,y]:

$$A[x,y] = 255 - \text{int}\{\ (255/64)(T[x,y] + ½)\ \}$$

The halftoning algorithm uses this array to compare with an input pixel I_i to binarize the output pixel I_o as follows:

$$\text{if } I_i \geq A[x,y] \quad I_o = 1; \quad \text{else } I_o = 0$$

For areas of constant pixel values in the input image, the 4x4 highlight cells will take on the shapes shown in Figure 2(a). The “size” of the center cluster, i.e., the number of white pixels that comprise the cluster, is indicated next to each cell. Likewise, Figure 2(b) lists all 16 shadow cells with the number of black cluster pixels. This list of cell shapes is of course dictated by the specification of the dither template in Figure 1.

A common misunderstanding is that these cell shapes are simply used as symbols that replace a gray level value in the image. It is important to note that when the input image contains high frequency detail that passes through a 4x4 cell, the shape of the cell will not appear as shown in these figures. The deviation of these shapes in response to edge detail is referred to in the industry as “partial dots” and is important for more accurate rendering.

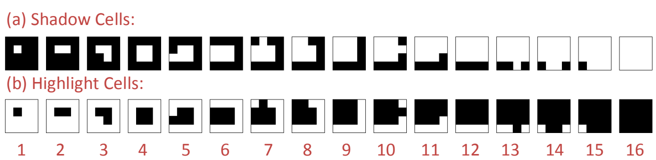

Figure 2. White and black clusters for (a) shadow cells and (b) highlight cells. The sizes of the clusters (in pixels) are also indicated.

Carrier Cells

In our solution, information will be embedded into a subset of these cells by single pixel shifts. Note that not all clusters have the same degree of freedom to move. We define the cells that have room to move to at least 2 single pixel shift positions “carrier cells”. In Figure 2, cells with size 1 through 9 are potential carrier cells. Cells 1 though 4 can move to all 8 surrounding positions and can thus carry 3 bits. Cells 5 and 6 can move to 4 positions and carry 2 bits. Cells 7, 8 and 9 can move to 2 positions and thus can only carry 1 bit.

The mapping of 1-, 2-, and 3-bit codes to shift positions are defined by a Shift Rule as shown in Figure 3. The unshifted position is shown in the center of each 3x3 array. An entry of “x” is used to indicate shift positions that are not used, and non-zero entries indicate the shift position of a code of that particular value. For example, for the 1-bit carrier shift rule, embedding a code of “0” will cause a shift right by 1 pixels and a shift up by 0 pixels; embedding a code of “1” will cause a shift right by 0 pixels and a shift down by 1 pixel. Sample shift positions for 1-bit, 2-bit and 3-bit carriers are illustrated in Figure 4.

1-bit carrier

x	x	x
x		0
x	1	x

2-bit carrier

x	0	x
x		1
x	3	2

3-bit carrier

0	1	2
7		3
6	5	4

Figure 3. Carrier Shift Rule.

In each of the 1-, 2-, and 3-bit cases of enabled carrier cells an unshifted carrier is reserved for a special symbol: a marker cell. Marker cells are an important means to communicate special signals to the recovery system. Several consecutive marker cells can be used, for instance, to indicate the separation between separate streams of payload bits.

While highlight and shadow cells 1 through 9 have the potential to move in directions as described above, it may not be possible to detect them all reliably during recovery. The viability of a particularly sized carrier cell to be detected at each of its candidate shift positions must be evaluated for the targeted printer. Knowledge of the recovery rates for each highlight and shadow cell must be incorporated in the encoding process. This goal is realized by a Selection Rule consisting of a table that specifies which carrier cells will be used, along with the bit-carrying capacity of those cells.

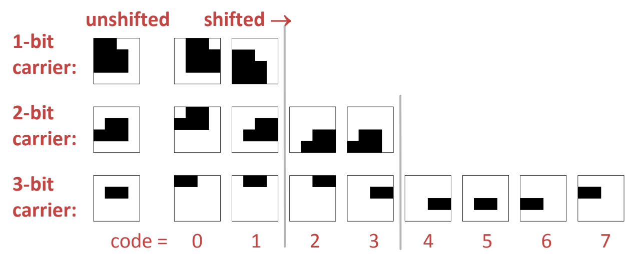

Figure 4. Shift positions for 1-, 2-, and 3-bit codes as defined by the Carrier Shift Rule.

Stegatone Generation System

The continuous-tone input image is referred to as the “Mule” image because it will carry the hidden data. The output is a steganographic halftone, or “stegatone”, that is targeted for a specific type of printer. A special subsystem called the Reference Image Generator is central to stegatone creation, and will be made available for the recovery system. The principal outputs are the Reference Halftone, and a Reference Map. The Reference Halftone is the standard clustered-dot halftone of the Mule identical to the Stegatone, but with all dot clusters unshifted. The Reference Map defines the bit-carrying capacity of each cell in the halftone. The input image will be segmented into blocks that correspond to halftone cluster locations. In the examples used in this document, halftone cells are comprised of 4 by 4 pixels. These cells are temporarily flattened by having all 16 pixels set to a value equal to their average.

The Reference Map Generator

The Reference Map has one pixel value for each halftone cell region, and is thus 1/4 the size of the input image in each dimension. It is this map that segments all 4x4 cells in the image as either carrier cells or non-carrier cells. Non-carrier cells are also called “reference cells” because they do not contain shifted clusters and can be used as anchors by the recovery system to establish the cell boundaries across the Stegatone. The value of pixels in the Reference Map are equal to the number of bits that cell is capable of carrying. Carrier cells have a value of 1, 2 or 3. A value of 0 indicates a reference cell. It should be noted that the Reference Map compresses to a size much smaller than the Mule since it has on ly 1/16 the number of values and only 2 bits per value.

The Reference Map is derived from the input image, with all 4x4 cells set to the average of that cell, and the carrier cell selection rule described earlier. With reference to the Dither Template (Figure 1), the image is first segmented into highlight and shadow cell regions. The Reference Map is populated by matching the cell average value to the number of bits defined by the carrier section rule. In Figure 5(a), a portion of an example Mule image in shown enlarged 3 times with its corresponding Reference Map in Figure 5(b). For the purpose of illustration, 1 bit carriers are shown in blue, 2 bit carriers in green, 3 bit carriers in red, and reference cells are shown in black. The Reference Map Generator also delivers a Carrier Count. The Carrier Count is the total bit capacity of all carrier cells in the Mule Image. For the Mule Image and Selection Rule in our example, the carrier count is 18,961 bits. The 600x600 pixel (1x1 inch) image has a total of 11,250 potential carrier cells of which 8,660 (or 77%) are identified as Carrier Cells.

It is important to note that only the cells identified as carriers in the original image are flattened (by replacing the cell with its average). The reference (non-carrier) cells areas are left intact for image quality reasons. These cells will retain the untouched high frequency detail that will be manifested as partial dots in the output halftone. The Reference Image Generator subsystem uses the Reference Map to selectively flatten cells before creating the Reference Halftone. For our example Mule image the Reference Halftone Image is shown in Figure 5(c).

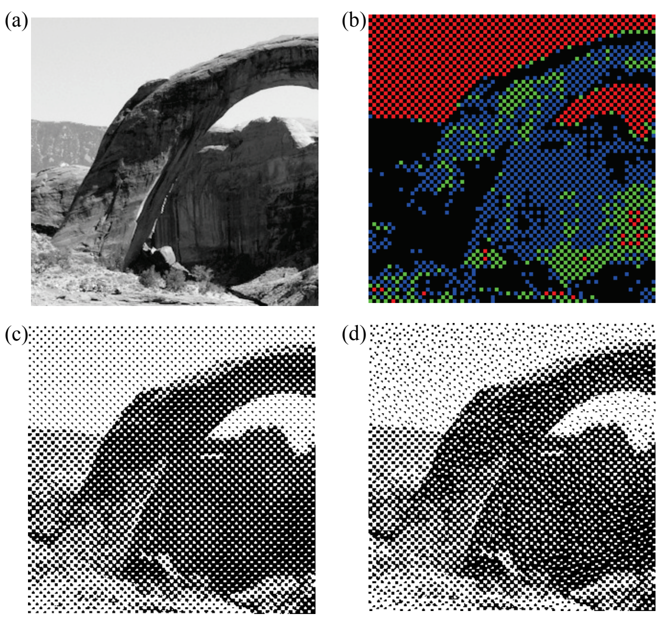

Figure 5. Enlarged portion of an example encoding. (a) Input "Mule" Image, (b) Reference Map, (c) Reference Halftone, (d) Stegatone.

Encoding the Payload

The Payload is the digital data to be embedded in the stegatone. Recovering from errors introduced into digital data encoded in halftone clusters can be thought of as similar to the process of reading a Compact Disc (CD). When decoding a CD, various types of redundancy and auxiliary information are used not only to compensate for the uncertainties associated with reading the physical pits in the media, but also to handle anticipated scratches and other degradations. Similarly, Stegatone prints are affected by uncertainties introduced in the process of printing and re-acquiring the image, which induces degradations such as smudges and other deformations caused by effects such as ink splatter, paper motion and optical blur.

The Carrier Count that is delivered from the Reference Map Generator for a given Mule Image and Carrier Selection Rule is an important piece of information to determine the size of Payload that can be accommodated. For Payload bit counts that are shorter than the Carrier Count, the Payload can be simply repeated over and over until all carrier cells are used. Alternatively, error correction codes can fill the available carrying capacity. A Shift List Generator uses the Reference Map to determine the order of Carrier Cells and their associated capacities. It separates the Payload into 1-, 2-, or 3-bit pieces according to the current carrier cell. The up-down, left-right shift is specified by the Carrier Shift Rule (Figure 3) for the number of bits used. The result is a shift list.

A Cell Shifter takes as input the Reference Halftone Image and generates the Stegatone. Special markers are indicated by unshifted carrier cells to assist recovery as mentioned above. The example Stegatone in Figure 5(d) used 3 marker cells to separate repeating instances of the Payload. Note that the small perturbations due to the encoding shifts impart a blue-noise-like visual appearance to the resulting rendering.

Printer Calibration and Recovery Rates

To determine the Carrier Selection Rule for a target printer, it is necessary to quantify the recovery rate for each type of cell, since not all carriers will survive the print-scan process. For this purpose, we use a special Mule Image that can uniformly test all candidate carrier cells. A non-trivial problem was devising a single repeating payload that would test all shift positions of each code. We want a string of bits that when separated into 3-, 2-, or 1-bit parts represents each possible code an equal number of times. This property is desirable so that the resulting recovery statistics are not biased by more of one type of code than another. Having a payload with this property allows for all codes to be tested in parallel, thus simplifying the calibration procedure. We found a solution for this problem and used the associated payload for our measurements. This payload was embedded in our special Mule Image to create a test stegatone which we printed on several different printers, then scanned the results.

For proper recovery, alignment is a central problem. Since we are measuring single pixel shifts, the scans must be de-skewed, scaled, and offset with sub-pixel accuracy to align the grid of cell boundaries. We found that every printer has slightly different horizontal and vertical pixel periods that must be addressed. Also, correction for "shearing" where the pixel grid deviates slightly from orthogonal must be performed in some cases. We then use a mean-square-error-based best match approach to read carrier shifts. Figure 6 shows a magnified sample of the same tiny crop region from our test scans for 4 example printers, along with the unshifted reference halftone and source stegatone used for testing. These images are part of the full test stegatone where each carrier type is shifted 300 times. Alignment grid lines are overlaid to show the 4x4 cells.

Figure 7 shows the results for these printers where recovery rates are expressed using the percentage of correctly decoded bits for each highlight (H) and Shadow (S) carrier cell size. This data helps us to reject the use of carriers that perform badly; in the figure, rejected carriers with poor recovery are indicated by a gray background. In all cases, these cells were small shadow cells where the tiny white shadow dots were consumed by the surrounding dot gain and rendered incapable of detection. Eliminating these poor performers, we then calculate the aggregate recovery rate for a printer by weighting the rate for each cell by the number of bits that cell can carry. Aggregate recovery rates for these printers are as follows:

B&W Laser	96.5%
Color Laser	97.9%
Inkjet	95.8%
Indigo 7000	99.4%

The results suggest that clustered dot halftones demonstrate enough robustness to indeed carry significant payloads. The per-cell recovery rates, combined with appropriate error correction coding, can ensure a reliable mechanism to embed and recover data in hard copy documents without unsightly bar codes.

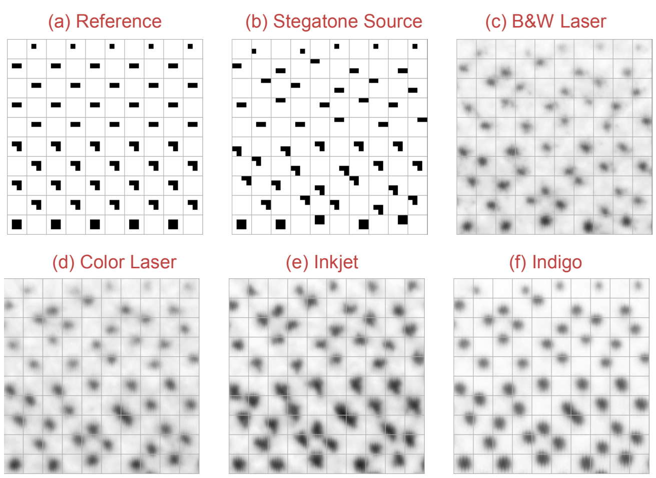

Figure 6. Tiny portion of scans of printed calibration stegatones on various printers enlarged at 12.5x.

Printer	H1	H2	H3	H4	H5	H6	H7	H8	H9
B&W Laser	91.8	97.1	97.3	97.4	90.5	95.8	99.3	99.3	99.3
Color Laser	93.3	97.8	99.2	99.8	100	98.7	100	100	100
Inkjet	91.3	93.2	92.9	97.0	95.0	98.2	98.7	98.7	99.3
Indigo	98.3	100	100	100	100	100	100	99.7	100

Printer	S1	S2	S3	S4	S5	S6	S7	S8	S9
B&W Laser	56.3	90.7	94.6	94.2	97.8	96.8	99.7	99.3	99.0
Color Laser	42.9	56.4	73.0	90.1	95.5	96.5	98.0	100	99.7
Inkjet	48.4	56.8	76.8	85.0	82.7	93.0	95.7	96.7	96.3
Indigo	39.2	47.1	67.2	97.0	97.5	99.2	99.7	100	99.7

Figure 7. Measured recovery rates for each Highlight (H) and Shadow (S) carrier for 4 sample printers. Gray boxes indicate rejected carriers.

References

[1] D. Kacker and J. P. Allebach, " Joint halftoning and watermarking," IEEE Trans. Signal Processing, vol. 51, no. 4, pp. 1054 –1068, Apr. 2003.

[2] P. Chiang, N. Khanna, A. Mikkilineni, M. Segovia, S. Suh, J. Allebach, G. Chiu, and E. Delp, "Printer and Scanner Forensics", IEEE Signal Processing Magazine, March 2009.

[3] N. Damera-Venkata, J. Yen, V. Monga, B. Evans, "Hardcopy Image Barcodes Via Block-Error Diffusion", IEEE Transactions On Image Processing, Vol. 14, No. 12, pp. 1977-1989, 2005.

[4] S. Yip; O.Au, C. Ho, H. Wong, "PI-preserve data hiding for halftone image", Proceedings of International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005, pp. 125 – 128, 13-16 Dec. 2005.

[5] A. Attar, O. Taheri, S. Sadri, R. Fattahi, "Data Hiding in Halftone Images using Error Diffusion Halftoning with Adaptive Thresholding", Canadian Conference on Electrical and Computer Engineering, CCECE '06, pp. 2029 – 2032, May 2006.

[6] M. Fu; O. Au, "Data hiding in halftone images by conjugate error diffusion", Proceedings of the International Symposium on Circuits and Systems, ISCAS '03, Vol. 2, pp. II-920 - II-923, May 2003.

[7] Wang, S., "Digital watermarking using conjugate halftone screens", US Patent 5,790,703, 1998.

[8] S. Pei; J. Guo, "Novel Data Hiding and Blind Decoding in Dithering Halftone Images", Fifth International Conference on Information, Communications and Signal Processing, pp. 1006-1009, 2005.

[9] Fu, M.S. Au, O.C., "Data hiding watermarking for halftone images", IEEE Tran. on Image Processing, Vol. II, pp. 477-484, April 2002.

[10] S. Suh, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, " Printer mechanism-level information embedding and extraction for halftone documents: New results," in Proc. IS&T's NIP23: Int. Conf. Digital Printing Technologies, Anchorage, AK, Sept. 2007, pp. 549 – 553.

[11] O. Bulan, V. Monga, G. Sharma, and B. Oztan, " Data embedding in hardcopy images via halftone-dot orientation modulation," in Proc. SPIE Int. Conf. Security, Steganography, and Watermarking of Multimedia Contents X, vol. 6819, pp. 68190C-1-12, Jan. 2008.

[12] B. Oztan and G. Sharma, "Continuous phase modulated halftones and their application to halftone data embedding," in Proc. IEEE Intl. Conf. Acoustics Speech and Sig. Proc., vol. II, May 2006, pp. 333–336.

[13] B. Oztan and G. Sharma, "Clustered-Dot Color Halftone Watermarks", in Proc. IS&T/SID Color Imaging Conference, pp 99-108, 2008.

[14] Pettersson, et al, "Coding pattern and apparatus and method for determining a value of at least one mark of a coding pattern", US Patent 6,663,008, 2003.

Author Biography

Robert Ulichney received a Ph.D. from MIT in electrical engineering and computer science. He was with Digital Equipment Corp. for several years where he focused on image and video implementations for both hard copy and display products. From 1998-2002 he was with Compaq's Cambridge Research Lab where he led a number of research efforts in video and image processing. Bob is now a Distinguished Technologist with HP Labs where his research activities include image enhancement, camera-projector systems and steganographic halftoning.